

## BAB IV

### PERANCANGAN DAN IMPLEMENTASI

Dalam bab perancangan ini akan dijelaskan secara rinci rancangan pengukuran daya komputasi melalui pendekatan metode *full virtualization* dan *paravirtualization*. Beberapa hal yang dibahas meliputi analisis kebutuhan, desain dan rancangan sistem.

#### 4.1 Analisis Kebutuhan

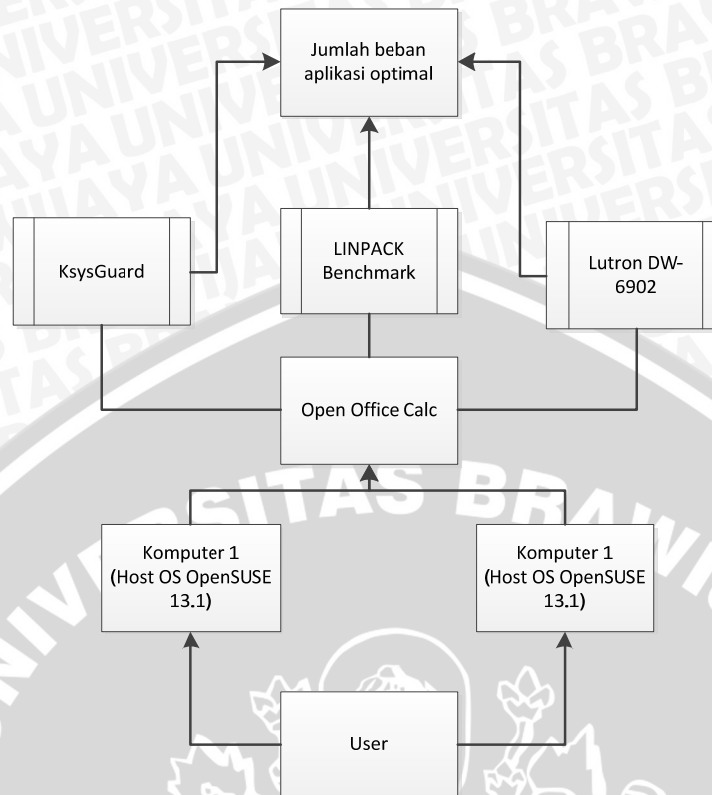
Pada analisis kebutuhan, akan dibahas komponen-komponen yang digunakan untuk melakukan pengukuran pada metode yang digunakan. Komponen-komponen yang akan dibahas meliputi spesifikasi *software* dan *hardware* yang digunakan.

Pada analisis kebutuhan akan dibedakan menjadi dua, sistematika pengukuran dan deskripsi mengenai metode pengukuran yang akan diimplementasikan.

##### 4.1.1 Sistematika Pengukuran

Untuk menghasilkan hasil analisa yang dapat menjawab rumusan masalah yang telah disusun pada bab sebelumnya, terlebih dahulu harus dibuat dahulu sistematika pengukuran yang tepat. Sistematika tersebut dapat dituangkan dalam bentuk diagram alur pengujian.

Dalam diagram yang akan dibuat, *user* (pengguna) perangkat dibagi menjadi 2 yaitu *user* (pengguna) perangkat secara konvensional dan *user* (pengguna) perangkat pada metode virtualisasi perangkat fisik dilakukan. Untuk lebih jelasnya dapat dilihat pada gambar 4.1



**Gambar 4.1** Penggunaan komputer oleh user pada metode konvensional

Sumber : perancangan

User pada kondisi di atas hanyalah sebagai pengguna komputer biasa pada umumnya, menjalankan sebuah komputer dengan beban aplikasi yang terdapat di dalam sistem operasi. Pengukuran akan dilakukan pada 3 kondisi berbeda, yaitu awal, *idle*, dan saat beban berjalan.

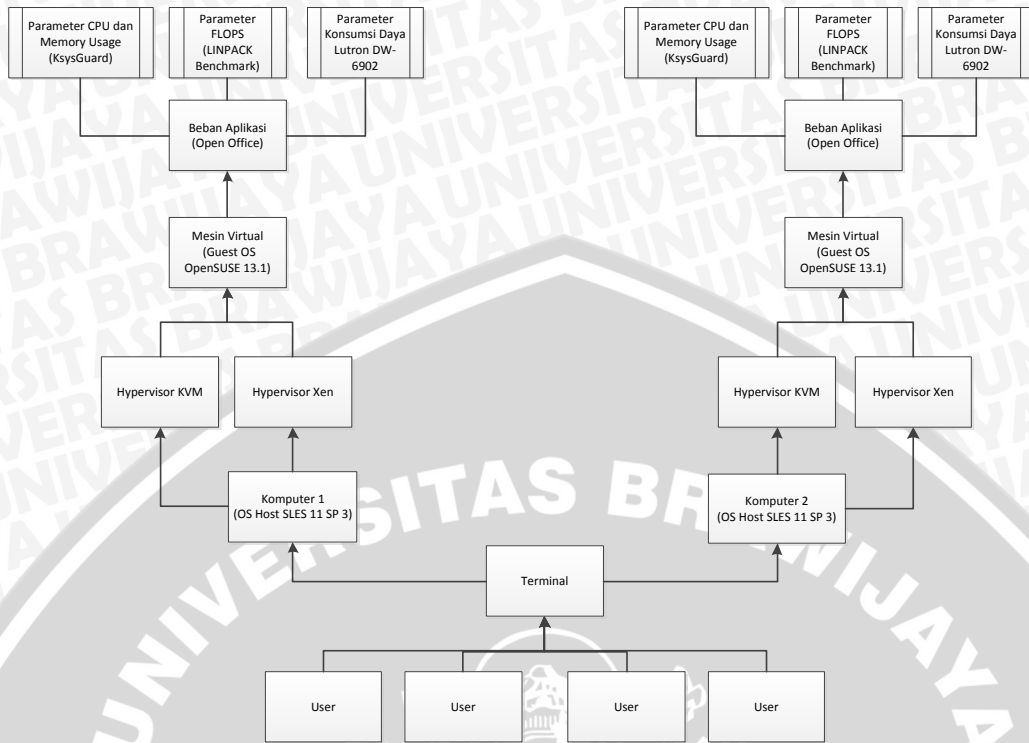
Sesuai diagram yang tergambar di atas, tahapan pengukuran yang akan dilakukan pada metode ini adalah :

- 1) User mengoperasikan komputer yang telah dilakukan instalasi OpenSUSE 13.1 sebagai sistem operasi utama.
- 2) User melakukan instalasi aplikasi perkantoran dan memodifikasi aplikasi Open Office Calc untuk dijalankan sebagai beban proses komputasi.
- 3) Pada kurun waktu dijelankannya beban aplikasi Open Office Calc tersebut, dilakukan monitoring *cpu usage* dan *memory usage* dan hasilnya direkam melalui aplikasi KsysGuard.

- 4) Penggunaan memori oleh beban aplikasi yang tercatat dalam KsysGuard, digunakan sebagai dasar pengukuran kemampuan komputasi pada *Linpack*.
- 5) Selanjutnya ketika pengukuran komputasi lewat *Linpack* dijalankan, maka bersamaan dengan itu diamati pula nilai konsumsi daya (*watt*) yang terlihat pada *wattmeter* Lutron DW-6092.
- 6) Jika penggunaan memori (*memory usage*) masih belum optimal, maka pemberian beban proses komputasi baru dapat diberikan.
- 7) Salin file Open Office Calc yang sebelumnya dijalankan menjadi file baru.
- 8) Dengan kondisi dokumen Open Office Calc sebelumnya masih berjalan, jalankan salinan file Open Office Calc yang baru.
- 9) Ulangi langkah 2) sampai 8) hingga didapati kondisi pada desktop terjadi gangguan baik berupa aplikasi yang berjalan tidak merespon perintah user (*hang*) maupun terpaksa dilakukan penghentian secara paksa (*kill*) untuk proses yang sedang berjalan.

Kemudian selanjutnya adalah pengukuran melalui metode kedua yang juga metode lanjutan dari metode sebelumnya, yaitu melalui penambahan metode virtualisasi perangkat fisik yang akan diuji. Sama halnya dengan metode konvensional, pengukuran akan dilakukan pada 3 kondisi berbeda, yaitu awal, *idle*, dan saat beban berjalan.

User pada metode yang kedua ini berperan sebagai *client* yang melakukan akses ke sebuah komputer yang difungsikan server mesin virtual. Meskipun pada kenyataannya pengukuran hanya berlangsung pada sisi server, secara skema diagram tetap harus menggambarkan adanya *user* yang mengakses server mesin virtual. Berbeda dengan *user* pada metode sebelumnya dimana sebuah komputer hanya dijalankan oleh seorang *user*, pada metode kedua akan ada kemungkinan sebuah komputer dapat diakses oleh banyak *user* dengan dijalankan mesin virtual di dalamnya. Untuk lebih detilnya, diagram dapat dilihat pada gambar 4.2



**Gambar 4.2** Penggunaan komputer oleh user pada metode virtualisasi perangkat fisik

Sumber : perancangan

Sesuai diagram yang digambarkan diatas, tahapan pengukuran yang akan dilakukan adalah sebagai berikut :

- 1) Komputer 1 dan 2 diberikan perlakuan yang sama pada tahapan pengukuran parameternya.
- 2) Pada masing-masing komputer, dilakukan instalasi SLES 11 SP 3 64-bit sebagai sistem operasi pada mesin *host*.
- 3) Setelah sistem operasi pada mesin *host* terinstall, dilakukan instalasi hypervisor KVM dan Xen.
- 4) Pada masing-masing hypervisor baik KVM maupun Xen, ditambahkan sebuah mesin *virtual* dimana *guest* OS yang digunakan adalah OpenSUSE 13.1 32-bit.
- 5) Kemudian di dalam mesin virtual yang telah dibuat, dijalankan beban aplikasi Open Office Calc berikut prosedur pengukuran yang sama diterapkan pada pengukuran metode konvensional.
- 6) Prosedur penambahan beban yang dilakukan dapat dihentikan jika pada *desktop* mesin virtual terjadi gangguan baik berupa aplikasi yang berjalan tidak merespon perintah user (*hang*) maupun terpaksa dilakukan penghentian secara paksa (*kill*) untuk proses yang sedang berjalan.

- 7) Setelah prosedur pengukuran untuk sebuah mesin virtual dilakukan, maka kondisi baru diterapkan melalui penambahan sebuah mesin virtual pada hypervisor KVM maupun Xen.
- 8) Pada masing-masing mesin virtual tersebut, dijalankan kembali beban proses komputasi berikut prosedur pengukurannya dan penambahan beban akan dihentikan sesuai kondisi yang dipaparkan sebelumnya.
- 9) Penambahan mesin virtual akan disesuaikan dengan spesifikasi dan kemampuan perangkat fisik mesin *host*.

#### 4.1.2 Spesifikasi Kebutuhan Sistem

Spesifikasi kebutuhan sistem akan membahas mengenai rincian komponen-komponen apa saja yang akan terlibat pada pengujian yang akan dilakukan baik komponen *software* maupun *hardware*.

##### 4.1.2.1 Spesifikasi Software

Untuk melakukan pengujian yang memadai, diperlukan dukungan perangkat lunak (*software*) sebagai berikut :

#### 1. Sistem Operasi Host

Pada penelitian ini, pengujian dilakukan dengan menggunakan sistem operasi SLES 11 Service Pack 3 yang dirilis pada tanggal 1 Juli 2013. SLES adalah sistem operasi berbasis Linux yang dikembangkan oleh SUSE, untuk difungsikan sebagai *server*, *mainframe*, dan *workstation*. Jika dibandingkan dengan openSUSE yang mengalami pembaharuan versi tiap 8 bulan, SLES dirilis tiap 3-4 tahun sekali untuk tiap versi, dengan pembaharuan *service pack* tiap 18 bulan sekali. Hal ini berkaitan dengan fungsi penggunaan SLES sebagai server sehingga diperlukan beragam pengujian produk yang lebih intens daripada openSUSE yang digunakan untuk kebutuhan personal.

Seperti halnya sistem operasi lain yang dikembangkan dalam distribusi SUSE, SLES 11 juga memiliki beberapa fitur yang serupa seperti berikut :

##### 1) YaST Control Center

SUSE menawarkan instalasi dan program administrasi yang bernama YaST2 untuk menangani masalah partisi hard disk, pengaturan sistem, RPM *package management*, pembaharuan sistem secara online, konfigurasi

jaringan dan *firewall*, serta pengelolaan pengguna. Dengan kata lain, YaST bisa disetarakan seperti Control Panel dalam sistem operasi Windows dan tidak dimiliki oleh distro Linux lainnya.

## 2) ZYpp

ZYpp atau Zypper adalah *software* dukungan di dalam YaST untuk memudahkan pengelolaan *package* baik berupa instalasi, akses *repository*, dan lain-lain.

## 3) Build Service

Open Build Service menyediakan *software* untuk melakukan *compile*, merilis dan mendistribusikan *software* mereka untuk distro Linux lainnya, termasuk Mandriva, Ubuntu, Fedora dan Debian. Open Build Service memberikan kemudahan dalam menyederhanakan proses *package* agar dapat berjalan dalam distribusi Linux lainnya dalam versi manapun yang digunakan, dengan memperbanyak versi *package* yang dapat didukung di atasnya.

## 4) Xgl dan Compiz

Xgl dan Compiz berperan dalam pemanfaatan kartu grafis melalui penyediaan driver OpenGL dan melakukan akselerasi pada OpenGL tersebut.

## 5) KDE dan GNOME

KDE adalah *software interface* grafis yang berjalan pada desktop untuk memudahkan pengguna untuk mengakses aplikasi yang ada di dalam SLES 11.

Dasar pemilihan sistem operasi SLES 11 Service Pack 3 untuk pengujian penelitian ini adalah :

### 1) Tepat untuk pengguna awam dengan kebutuhan jaringan dalam lingkup kecil.

Jika dibandingkan dengan distro Linux lainnya seperti Red Hat Enterprise Linux, Debian, CentOS yang secara umum difungsikan sebagai server, SLES 11 lebih tepat untuk dioperasikan oleh pengguna yang awam dalam mengoperasikan sistem berbasis Linux berkat dukungan *interface*

pada desktop yang memudahkan pengguna untuk mengakses aplikasi yang tersedia.

Selain itu dengan kemudahan untuk dapat dioperasikan pada komputer pribadi, SLES 11 dapat difungsikan sebagai *client* maupun server untuk kebutuhan konfigurasi jaringan lingkup kecil seperti LAN.

## 2) Dukungan aplikasi yang banyak

SUSE sebagai salah satu distro Linux yang mampu bertahan dengan mengembangkan sistem operasi dan software berbasis Linux yang beroperasi dalam sistem sendiri, tentunya memberikan perhatian lebih terkait dukungan aplikasi yang dapat dijalankan pada SLES 11. SLES 11 dapat menjalankan sekitar 6000 aplikasi resmi dari 1500 penyedia *software* mandiri, yang tentunya lebih banyak jika dibandingkan distro Linux lainnya.

## 3) Keamanan yang terjamin

Sebagai sistem operasi yang ditujukan untuk kepentingan komersil, faktor keamanan pada SLES 11 sangat diperhatikan oleh pihak SUSE, baik pada saat pengujian produk sebelum dirilis hingga selama layanan sistem operasi tersebut aktif.

Pengujian yang intens perlu dilakukan untuk menghasilkan sistem yang paling stabil termasuk dari segi keamanan. Kemudian setelah produk dirilis, SUSE akan menjamin keamanan fitur dan perbaikan bug melalui *update packages* yang dapat diunduh melalui YaST Online Update yang terdapat pada SLES 11.

## 4) Kompatibilitas terhadap perangkat fisik maupun virtual

SLES 11 dirancang untuk dapat berjalan pada hampir sebagian besar perangkat fisik desktop maupun server, yang di dalamnya termasuk Intel/AMD 32bit dan 64bit x86, Intel Itanium, IBM Power, dan System z. Selain itu, SLES 11 juga memiliki dukungan terhadap penggunaan hypervisor dalam virtualisasi perangkat keras seperti VMWare, KVM 1.4, dan Xen 4.2.

Sistem operasi SUSE Linux Enterprise Server (SLES) 11 SP 3 yang digunakan pada pengujian memiliki *kernel version* 3.0.101 dengan sistem operasi

bekerja pada sistem 64-bit. Pemilihan SLES 11 SP 3 64-bit sebagai sistem operasi mesin *host* dikarenakan instalasi KVM dan XEN tidak dapat dilakukan pada sistem berbasis 32-bit.

## 2. Sistem Operasi Metode Konvensional dan *Guest*

OpenSUSE 13.1 diperlukan sebagai sistem operasi utama pada pengujian metode konvensional dan sebagai sistem operasi *guest* pada metode virtualisasi perangkat fisik dengan *kernel version* 3.11.6. OpenSUSE 13.1 yang digunakan adalah 32-bit karena lebih ringan untuk dijalankan pada mesin virtual. SLES dan OpenSUSE memiliki kesamaan fitur antara satu dengan yang lain dan berasal dari satu basis distro Linux yaitu SUSE. Namun keduanya memiliki perbedaan fungsi penggunaan, yaitu SLES dikembangkan khusus untuk perangkat yang difungsikan sebagai server sehingga fitur keamanan sistem lebih lengkap tersedia. Sedangkan OpenSUSE adalah sistem operasi *open-source* yang pengembangannya ditujukan untuk penggunaan pribadi oleh pengguna sehingga memiliki keunggulan dari segi tampilan *desktop*.

## 3. Hypervisor

Ada 2 jenis hypervisor yang digunakan dalam pengujian melalui pendekatan metode *full virtualization* dan *paravirtualization*.

Hypervisor yang pertama adalah KVM untuk pengujian dengan metode *full virtualization*. KVM adalah produk hypervisor yang dirintis oleh innotek GmbH, yang kemudian dikembangkan oleh Oracle Corporation dan dapat dijalankan pada arsitektur komputer berbasis x86 dan AMD/Intel64. KVM sangat populer di kalangan pengguna hypervisor karena sifatnya yang *open source* dan gratis; antarmuka yang digunakan memiliki tampilan yang terbaik dengan kemudahan pengelolaan mesin *virtual* untuk pemula; mendukung sistem *cross platform* secara penuh sehingga sistem operasi Linux, Windows, dan UNIX dapat dijalankan baik pada mesin *host* maupun *guest*; dan sudah terintegrasi untuk berjalan pada sistem operasi berbasis SUSE.

Hypervisor yang kedua adalah Xen yang digunakan dalam pengujian dengan metode *paravirtualization*. Xen dikembangkan oleh Xen Project untuk membuat hypervisor yang mampu menjalankan beragam sistem operasi *guest* pada mesin



virtual secara bersamaan pada sebuah mesin *host*. Hal ini dikarenakan mesin virtual sadar bahwa sedang divirtualisasikan sehingga tidak membutuhkan penambahan perangkat virtual, dan sebagai gantinya mesin virtual tersebut akan memberikan panggilan khusus melalui Xen Project untuk mengakses CPU, *storage*, maupun *network* yang terdapat pada mesin *host*.

**Tabel 4.1** Perbandingan kemampuan hypervisor

	Xen	KVM	VirtualBox	VMWare
Para-virtualization	Yes	No	No	No
Full virtualization	Yes	Yes	Yes	Yes
Host CPU	x86, x86-64, IA-64	x86, x86-64, IA64, PPC	x86, x86-64	x86, x86-64
Guest CPU	x86, x86-64, IA-64	x86, x86-64, IA64, PPC	x86, x86-64	x86, x86-64
Host OS	Linux, UNIX	Linux	Windows, Linux, UNIX	Proprietary UNIX
Guest OS	Linux, Windows, UNIX	Linux, Windows, UNIX	Linux, Windows, UNIX	Linux, Windows, UNIX
VT-x / AMD-v	Opt	Req	Opt	Opt
Cores supported	128	16	32	8
Memory supported	4TB	4TB	16GB	64GB
3D Acceleration	Xen-GL	VMGL	Open-GL	Open-GL, DirectX
Live Migration	Yes	Yes	Yes	Yes
License	GPL	GPL	GPL/proprietary	Proprietary

#### 4. File ISO

File ISO adalah file arsip dari media digital atau disk yang dibuat dengan menggunakan format konvensional diambil dari ISO sistem berkas. File yang tersimpan dalam bentuk ISO (ISO image) dapat dibuka, diedit, dienkripsi dan di-mount melalui *virtual drive*. Pada pengujian yang akan dilakukan, file ISO akan dibutuhkan pada saat instalasi sistem operasi *guest*.

##### 1) Instalasi sistem operasi *guest* pada KVM

Setelah proses instalasi hypervisor KVM selesai dilakukan, selanjutnya perlu dilakukan instalasi sistem operasi *guest* melalui penggunaan File ISO.

##### 2) Instalasi sistem operasi *guest* pada Xen.

Seperti halnya pada KVM, dilakukan instalasi juga pada sistem operasi *guest* OpenSUSE 13.1 untuk mendukung aplikasi Apache Open Office di dalamnya.

#### 5. Apache Open Office

Dengan meninjau pengumpulan data pada parameter yang dilakukan untuk kepentingan penyusunan hipotesis, tentunya pemilihan objek pengujian perlu disesuaikan juga dengan hasil kolektif data hipotesis yang telah dilakukan. Oleh karena itu Apache Open Office dipilih sebagai salah satu aplikasi yang sesuai dengan data yang telah diambil sebagai bahan analisis hipotesis. Selain itu, karena

sistem operasi yang digunakan berbasis Linux maka hal tersebut menjadi alasan lain digunakannya Apache Open Office sebagai beban proses yang akan dijalankan.

Apache Open Office itu sendiri merupakan sebuah paket aplikasi yang terdiri atas 6 komponen, yaitu Open Office Writer, Open Office Calc, Open Office Impress, Open Office Base, Open Office Draw, dan Open Office Math. Namun untuk pengujian yang akan dilakukan, hanya akan melibatkan Open Office Calc dengan pertimbangan bahwa aplikasi ini yang paling relevan sebagai objek pengukuran *floating point operation per second (flops)* dalam melakukan proses komputasi.

## 6. KsysGuard

KsysGuard adalah salah satu aplikasi Linux dengan fungsi untuk memantau penggunaan sumber daya cpu (*cpu usage*), memori (*memory usage*), dan lalu lintas paket data pada jaringan (*network history*). Fungsi lain yang dimiliki adalah pemantauan proses dan servis yang sedang berjalan serta alokasi sistem file pada harddisk yang digunakan. KsysGuard dapat dijalankan langsung tanpa melakukan instalasi *dependency* terlebih dahulu karena sudah otomatis terinstall secara *default* baik pada SLES 11 SP 3 maupun OpenSUSE 13.1.

Selain sebagai tools monitoring, KsysGuard memiliki kemampuan untuk merekam data yang dihasilkan dari proses *monitoring*.

## 7. Linpack Benchmark

Linpack *benchmark* adalah aplikasi yang memiliki fungsi untuk melakukan pengujian yang dikhususkan pada kemampuan prosesor dalam menghasilkan nilai *floating point operation per second (flops)*. Metode pengujian yang dipakai oleh Linpack benchmark adalah dengan memberikan sejumlah permasalahan matematis untuk diselesaikan oleh prosesor, berupa sistem persamaan linear  $Ax=b$ . Persamaan linear tersebut didefinisikan menjadi 4 bagian yang dapat diinput oleh user, yaitu :

- Jumlah persamaan yang akan diselesaikan (*problem size*)
- Jumlah baris array yang diketahui dan nilainya minimal harus sama dengan *problem size* yang dimasukkan.
- Jumlah tes pengukuran komputasi yang ingin dijalankan (*number of trials run*).

- Nilai keselarasan (*alignment value*). Nilai ini bergantung pada arsitektur sistem operasi yang digunakan. Jika sistem operasinya berbasis 32-bit maka nilai ini diisi dengan angka 4, sedangkan pada sistem operasi berbasis 64-bit maka diisi dengan angka 8.

Jika melihat parameter yang perlu dimasukkan oleh *user*, bisa dikatakan bahwa metode perhitungan yang dilakukan oleh Linpack adalah penyelesaian matriks dengan dimensi NxN dimana *problem size* mewakili koefisien matriks dan jumlah baris *array* sebagai konstanta matriks.

#### 4.1.2.2 Analisis Kebutuhan Hardware

Hardware yang digunakan untuk mendukung pengujian daya komputasi melalui pendekatan metode *full virtualization* dan *paravirtualization* antara lain : *personal computer* (PC) yang digunakan selayaknya dalam kondisi penggunaan konvensional dan kondisi sebagai mesin *host* sekaligus server mesin *virtual*, *network interface card* (NIC), kabel UTP, switch dan wattmeter.

##### 1. *Personal Computer* (PC)

Komputer yang digunakan sejumlah 2 unit dengan spesifikasi prosesor yang berbeda untuk nantinya dilakukan perbandingan kinerja prosesornya sesuai parameter yang diukur. Rincian spesifikasi komputer yang digunakan sebagai berikut :

Tabel 4.2 Spesifikasi PC yang diuji

Perangkat	Komputer 1	Komputer 2
Spesifikasi Prosesor	Intel Core i3-2120 (2 Cores, 3.3GHz)	Intel Core i3-3240 (2 Cores, 3.4GHz)
RAM	2 GB	2 GB
Disk	500 GB	500 GB

##### 2. *Network Interface Card* (NIC)

NIC merupakan perangkat pada *physical layer* pada OSI layer yang berfungsi untuk menghubungkan komputer dengan jaringan komputer, dimana

salah satu kegunaannya pada pengujian adalah untuk memperoleh akses internet yang dibutuhkan dalam mengakses repositori SLES dan OpenSUSE. NIC yang digunakan adalah pada komputer 1 adalah tipe Broadcom BCM95722 Gigabit Ethernet, sedangkan pada komputer 2 adalah tipe BCM5764M Gigabit Ethernet.

### 3. *Switch*

*Switch* merupakan perangkat pada layer *datalink*. *Switch* dapat menangani beberapa sambungan sekaligus pada saat yang sama. Tiap-tiap port 100Base-TX pada sebuah *switch* dapat mengirim dan menerima frame-frame secara bersamaan (*Full-Duplex*).

### 4. *Router*

*Router* menghubungkan antar jaringan yang berbeda segmen dan mengarahkan paket menuju tujuan sesuai informasi alamat IP.

### 5. *Wattmeter*

*Wattmeter* digunakan untuk mengukur konsumsi daya yang diperlukan oleh prosesor dalam melakukan proses komputasi. Untuk memperoleh nilai tersebut, daya akan diukur dalam 2 kondisi yaitu pada saat kondisi *idle* dimana tidak ada proses komputasi yang dilakukan prosesor, dan kemudian pada saat pengukuran kemampuan komputasi oleh Linpack Benchmark dijalankan.

*Wattmeter* yang akan digunakan adalah Lutron DW-6092 yang menyediakan pembacaan daya (KW), energi (KWh), arus (ampere) dan tegangan (volt). Hasil pembacaan oleh alat, dapat disimpan ke dalam SD Card dalam format dokumen .xls.



**Gambar 4.3** Wattmeter Lutron DW-6092

Sumber : **Lutron Electronic**

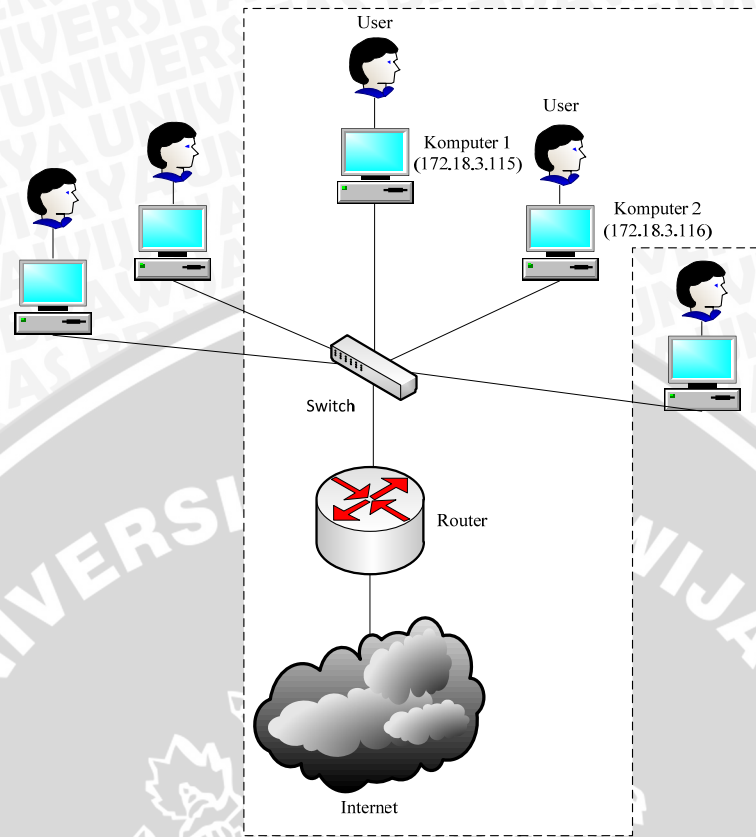
**Tabel 4.3** Tabel Spesifikasi Lutron DW-6092

(Sumber : Lutron Electronic)

Circuit	Custom one-chip of microprocessor LSI circuit				
Display	<ul style="list-style-type: none"> <li>* LCD Size : 81.4 X 61 mm ( 3.2 X 2.4 inch )</li> <li>* Dot Matrix LCD (320 X 240 pixels ) with back light.</li> </ul>				
Measurement	<ul style="list-style-type: none"> <li>* ACV</li> <li>* ACA</li> <li>* AC WATT ( True Power )</li> <li>AC WATT( Apparent Power )</li> <li>AC WATT( Reactive Power )</li> <li>* Power factor</li> <li>* Phase angle</li> <li>* Frequency</li> </ul>				
Wire connections	1P/2W, 1P/3W, 3P/3W, 3P/4W.				
Voltage ranges	10 ACV to 600 ACV, auto range.				
Current ranges	0.2 ACA to 1200 ACA, auto range/manual range.				
Safety standard	IEC1010 CAT III 600 V.				
ACV input impedance	10 Mega ohms.				
Range select	<table border="1"> <tr> <td>ACV</td> <td>Auto range.</td> </tr> <tr> <td>ACA</td> <td>Auto range &amp; manual range.</td> </tr> </table>	ACV	Auto range.	ACA	Auto range & manual range.
ACV	Auto range.				
ACA	Auto range & manual range.				
Clamp frequency response	40 Hz to 1 KHz.				
Spec. tested frequency	45 to 65 Hz.				
Over load protection	<table border="1"> <tr> <td>ACV</td> <td>720 ACV rms</td> </tr> <tr> <td>ACA</td> <td>1300 ACA with clamp probe CP-1200</td> </tr> </table>	ACV	720 ACV rms	ACA	1300 ACA with clamp probe CP-1200
ACV	720 ACV rms				
ACA	1300 ACA with clamp probe CP-1200				

#### 4.2 Desain Topologi Sistem

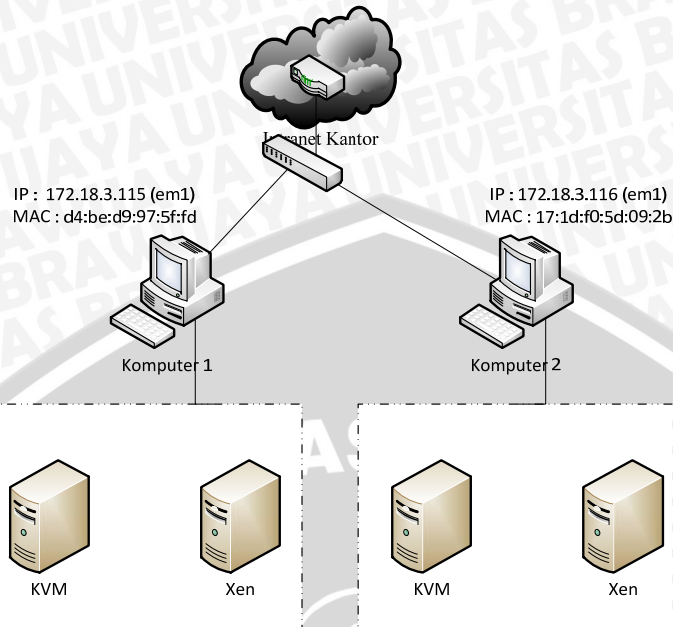
Pembahasan desain topologi sistem yang akan diuji mencakup tentang skema jaringan yang diterapkan pada objek pengujian, dan terbagi menjadi 2 (dua) bentuk topologi yaitu :



**Gambar 4.4** Topologi metode konvensional

Sumber : perancangan

Pada metode konvensional, sistem jaringan yang digunakan adalah jaringan LAN dengan konfigurasi jaringan yang mendukung adanya akses internet untuk kebutuhan akses *repository* pada *YasT Software Management*. Tentunya pada sistem jaringan LAN tersebut, tidak hanya terdapat satu komputer saja yang beroperasi tetapi dengan kondisi ada komputer lain yang terhubung juga pada perangkat *switch*. Namun untuk kepentingan pengujian, hanya akan ada 2 komputer yang digunakan sebagai objek pengujian (yang berada di dalam area garis putus-putus).



**Gambar 4.5** Topologi metode full virtualization dan paravirtualization

Sumber : perancangan

Pada pendekatan metode virtualisasi perangkat fisik (*full virtualization dan paravirtualization*), komputer 1 dan 2 akan difungsikan sebagai server mesin virtual yang nantinya dapat akses oleh *user*. Meskipun begitu, karena sifat pengukuran yang hanya akan dilakukan pada sisi server (komputer 1 dan 2) maka mesin virtual yang akan dibuat sudah dianggap telah diakses oleh banyak *user* sebagai *client* yang menggunakan perangkat fisik berkemampuan rendah. Dengan kata lain, perangkat fisik (komputer desktop) yang digunakan *client* tidak menjadi salah satu aspek kebutuhan yang harus tersedia.

#### 4.3 Implementasi Desain Pengujian

Dari desain topologi yang telah dibuat, berikutnya dilakukan implementasi sistem terkait instalasi dan konfigurasi yang dibutuhkan untuk mendukung berjalannya proses pengukuran daya komputasi yang akan dilakukan. Implementasi yang akan dilakukan terbagi menjadi dua, yaitu implementasi pengujian untuk metode konvensional dan implementasi pengujian untuk metode virtualisasi. Kedua implementasi desain tersebut akan diterapkan untuk masing-masing komputer

### 4.3.1 Implementasi Pengujian Metode Konvensional

Di dalam bagian ini, implementasi hanya terbatas pada desain untuk metode konvensional dimulai dari persiapan komponen lewat instalasi, konfigurasi, hingga tahapan proses pengukuran dapat dilakukan secara menyeluruh.

#### 4.3.1.1 Instalasi Sistem Operasi Utama

Pada metode konvensional, yang pertama kali perlu disiapkan tentunya instalasi sistem operasi utama. Sistem operasi utama yang dipakai adalah OpenSUSE 13.1 32-bit. Tahapan proses instalasi itu sendiri akan lebih lengkap dijelaskan pada bagian lampiran.

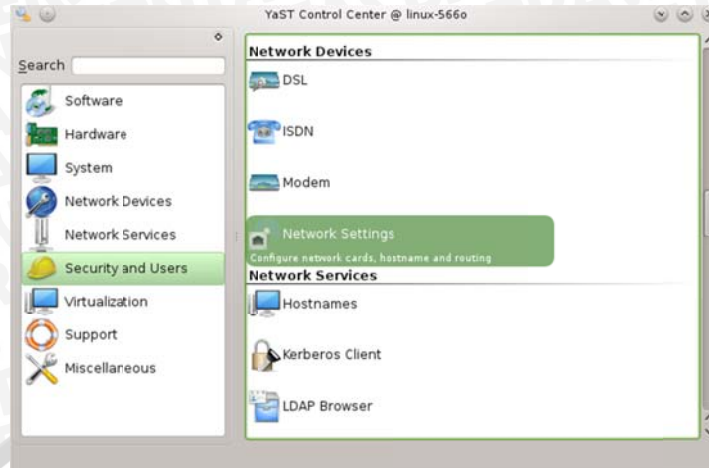


**Gambar 4.6** Tampilan desktop OpenSUSE 13.1

Sumber : implementasi

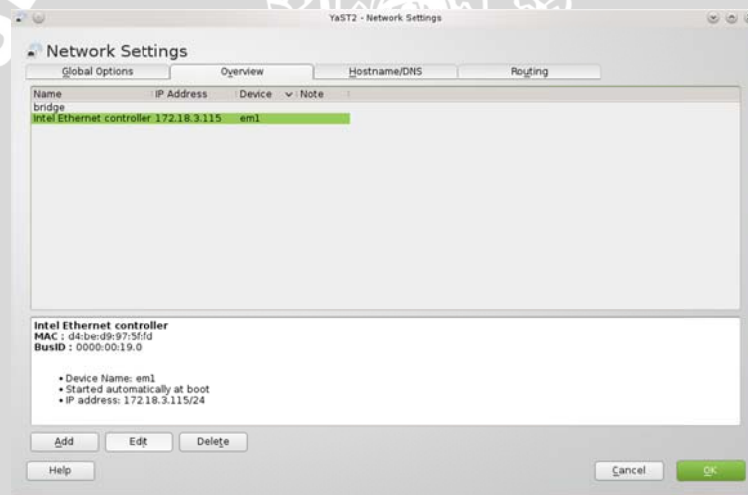
Setelah proses instalasi selesai, pemberian alamat IP (*ip address*) dilakukan guna mempermudah OpenSUSE untuk mengakses *online repositories*. Alamat IP yang digunakan adalah alamat intranet Unit Pernerangan Internasional berupa jaringan *local area* yang terkoneksi dengan jaringan internet. Konfigurasi alamat IP dilakukan pada *YasT->Network Setting*.





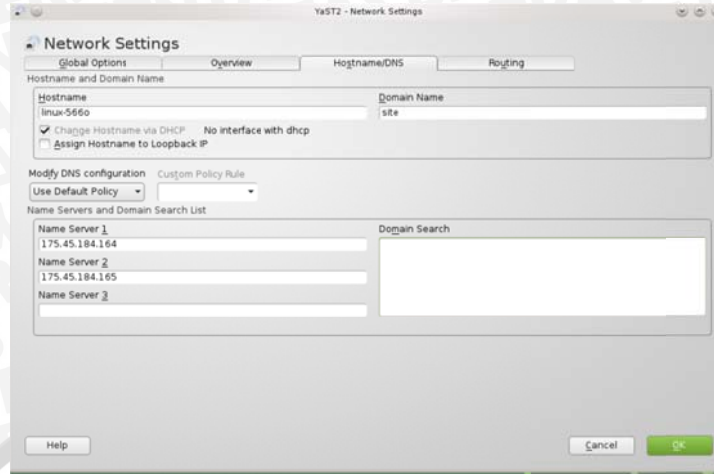
**Gambar 4.7** Tampilan menu konfigurasi jaringan pada YaST

Sumber : implementasi



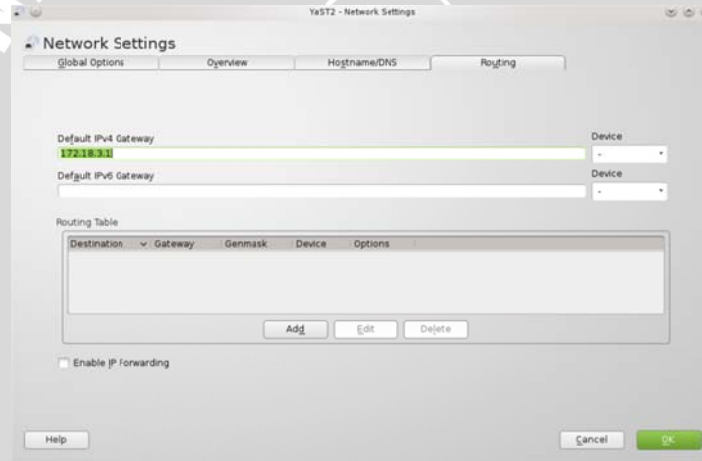
**Gambar 4.8** Tampilan konfigurasi alamat IP pada device em1

Sumber : implementasi



**Gambar 4.9** Tampilan konfigurasi *name server*

Sumber : implementasi



**Gambar 4.10** Tampilan konfigurasi *IPv4 Gateway*

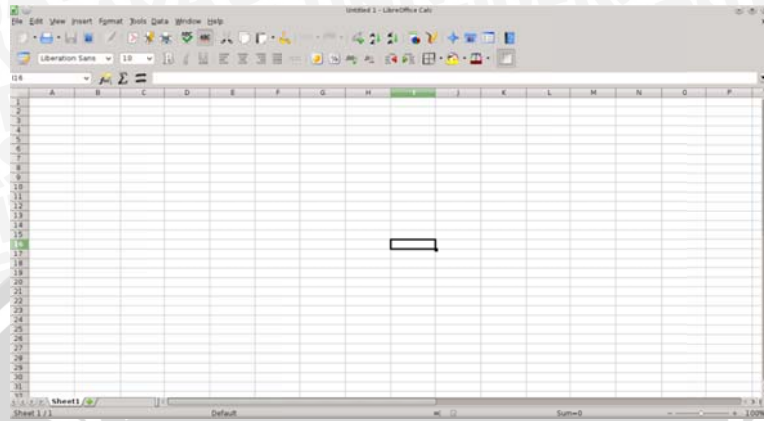
Sumber : implementasi

#### 4.3.1.2 Instalasi dan Konfigurasi Open Office Calc

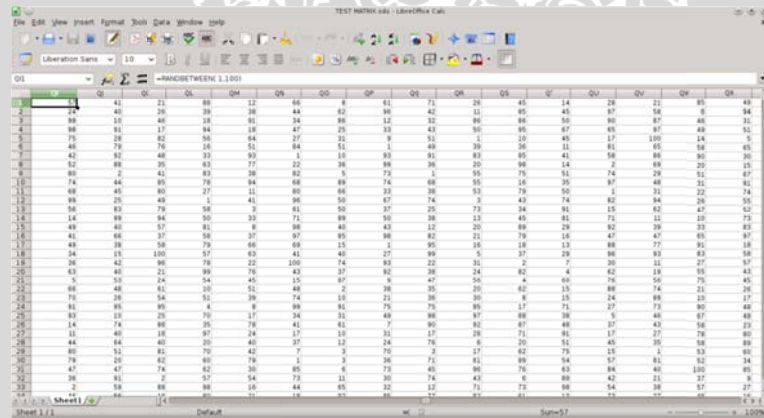
Seperti yang telah dijelaskan sebelumnya, OpenSUSE memiliki kemudahan dari segi pencarian *package* aplikasi. Sehingga dalam melakukan instalasi Open Office Calc, cukup dicari melalui *online repositories* lewat YaST-> Software Management. Di dalamnya akan dapat ditemukan *package* Open Office berikut *dependency* yang otomatis akan terunduh.

Setelah proses instalasi selesai, Open Office Calc dijadikan beban dengan memasukkan perhitungan perkalian matriks dengan dimensi besar, yaitu 1000x1000 untuk memperpanjang durasi berjalannya proses komputasi sekaligus memberikan

beban komputasi yang berat untuk mengetahui penggunaan memori yang dihasilkan. Nantinya dari masing-masing beban komputasi yang diberikan, akan muncul nilai kemampuan komputasi yang terbaik dari tiap-tiap beban.



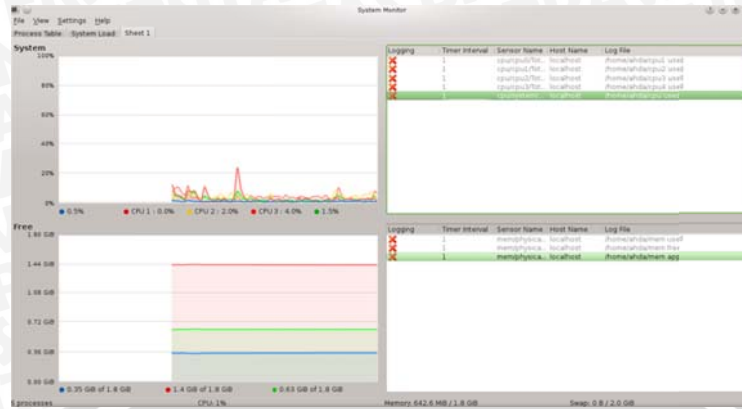
**Gambar 4.11** Tampilan Open Office Calc sebelum diberikan beban  
 Sumber : implementasi



**Gambar 4.12** Tampilan Open Office Calc saat dijadikan beban matriks  
 Sumber : implementasi

### 4.3.1.3 Monitoring Menggunakan KsysGuard

Untuk merekam data monitoring selama beban proses Open Office Calc berjalan, digunakan KsysGuard yang sudah terinstall secara *default* di dalam sistem operasi OpenSUSE 13.1. Pada KsysGuard, terdapat bagian yang dapat dimodifikasi oleh *user* untuk melakukan pemantauan terhadap sensor perangkat fisik yang dimiliki oleh komputer. Sensor yang akan dipantau adalah *cpu usage* dan *memory usage*.



**Gambar 4.13** Tampilan KsysGuard dalam melakukan proses *monitoring* dan merekam data dari *sensor cpu usage* dan *memory usage*

Sumber : implementasi

#### 4.3.1.4 Pengukuran Komputasi Menggunakan Linpack Benchmark

Dari penggunaan memori (*memory usage*) yang terekam melalui KsysGuard, nilai tersebut akan digunakan sebagai acuan untuk menentukan dimensi pengukuran yang akan diperoleh. Hal ini dimaksudkan untuk mencari nilai komputasi dari penggunaan memori (*memory usage*) oleh beban proses komputasi.

```

ahda: bash - Konsole
File Edit View Bookmarks Settings Help
Number of equations to solve (problem size): 1000
Leading dimension of array: 1000
Number of trials to run: 5
Data alignment value (in Kbytes): 4
Current date/time: Wed Jul 23 17:45:13 2014

CPU frequency: 3.291 GHz
Number of CPUs: 1
Number of cores: 2
Number of threads: 4

Parameters are set to:
Number of tests: 1
Number of equations to solve (problem size) : 1000
Leading dimension of array : 1000
Number of trials to run : 5
Data alignment value (in Kbytes) : 4

Maximum memory requested that can be used=8024096, at the size=1000

===== Timing linear equation system solver =====
Size LDA Align. Time(s) GFlops Residual Residual(norm) Check
1000 1000 4 0.057 11.7464 1.029949e-12 3.510325e-02 pass
1000 1000 4 0.039 16.9799 1.029949e-12 3.510325e-02 pass
1000 1000 4 0.039 16.9774 1.029949e-12 3.510325e-02 pass
1000 1000 4 0.039 16.9615 1.029949e-12 3.510325e-02 pass
1000 1000 4 0.039 17.1013 1.029949e-12 3.510325e-02 pass

Performance Summary (GFlops)
Size LDA Align. Average Maximal
1000 1000 4 15.9533 17.1013

Residual checks PASSED

End of tests

linux-5660: ~ #
ahda: bash
    
```

**Gambar 4.14** Tampilan Linpack Benchmark yang dijalankan pada terminal *super user*

Sumber : implementasi

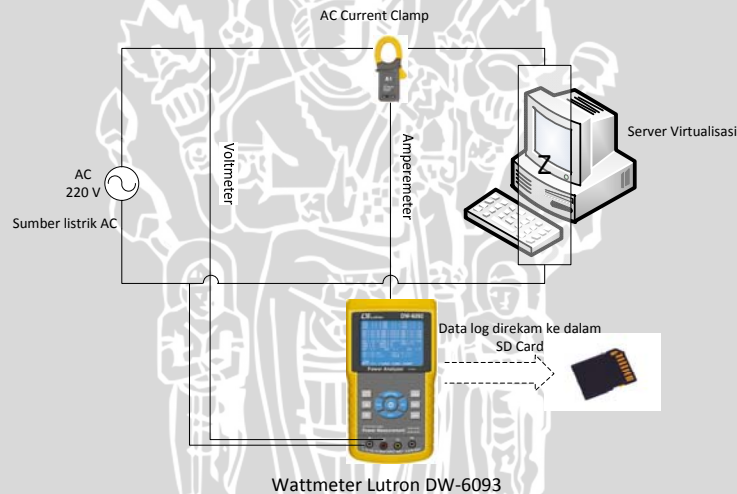
#### 4.3.1.5 Pengukuran Konsumsi Daya Menggunakan Wattmeter Lutron DW-6902

Pengukuran konsumsi daya dilakukan pada 2 kondisi, yaitu pada saat *idle* dan saat proses pengukuran komputasi *Linpack Benchmark* berjalan. Untuk mencari konsumsi daya listrik oleh prosesor dalam melakukan komputasi, diperoleh dari rumus perhitungan :

$$P \text{ komputasi} = P \text{ linpack} - P \text{ idle}$$

Dimana :

- *P komputasi* : konsumsi daya listrik yang digunakan saat proses komputasi
- *P linpack* : daya listrik yang terukur saat *Linpack Benchmark* berjalan
- *P idle* : daya listrik yang terukur pada saat kondisi *idle*.



**Gambar 4.15** Skema pengukuran daya listrik menggunakan wattmeter Lutron DW-6092

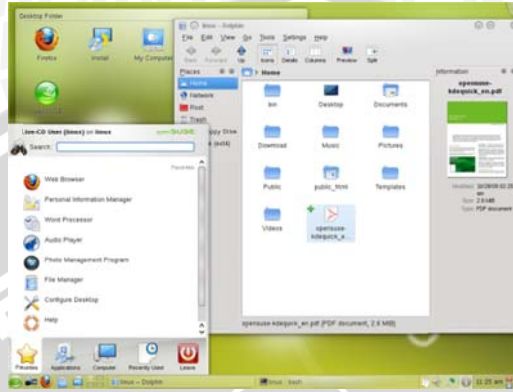
Sumber : implementasi

#### 4.3.2 Implementasi Pengujian Metode Virtualisasi

Pada metode virtualisasi, instalasi dan konfigurasi baru yang perlu ditambahkan hanya diterapkan pada sisi sistem *host*. Sedangkan pada sistem *guest*, tahapan yang dilalui pada tiap-tiap penambahan mesin virtual merupakan perulangan tahapan dari implementasi pengujian dengan metode konvensional.

### 4.3.2.1 Instalasi Sistem Operasi Host

Sistem operasi untuk *host (host OS)* yang digunakan adalah SUSE Linux Enterprise 11 SP 3 64-bit. Untuk proses instalasi sistem operasi yang dilakukan, dapat dilihat secara rinci pada bagian lampiran.



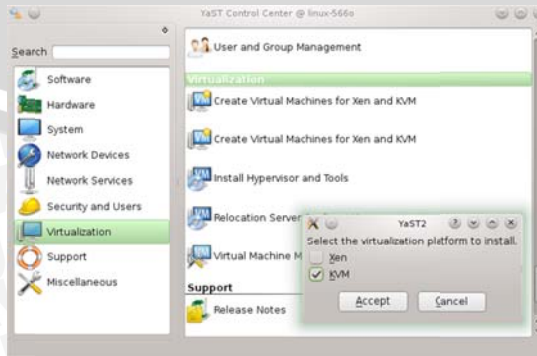
**Gambar 4.16** Tampilan desktop SLES 11 SP 3

Sumber : implementasi

Pada gambar 4.15, terlihat bahwa tampilan desktop pada SLES 11 SP 3 tidak jauh berbeda dengan tampilan desktop pada OpenSUSE 13.1. Sama halnya pada metode konvensional, setelah proses instalasi sistem operasi SLES selesai maka dilakukan pengalamanan IP guna mengakses *online repositories* dengan prosedur konfigurasi yang sama saat menggunakan OpenSUSE 13.1.

### 4.3.2.2 Instalasi KVM Untuk *Full Virtualization*

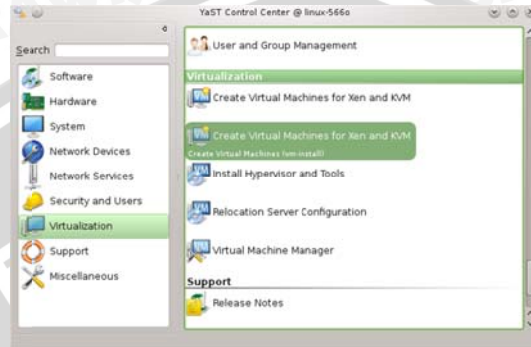
Untuk melakukan instalasi KVM, tahapan pertama yang dilakukan adalah melakukan instalasi hypervisor yang tersedia pada SUSE Linux Enterprise 11 SP 3 melalui menu *Yast -> Virtualization -> Install Hypervisors and Tools*.



**Gambar 4.17** Instalasi hypervisor KVM pada YasT sistem operasi *host*

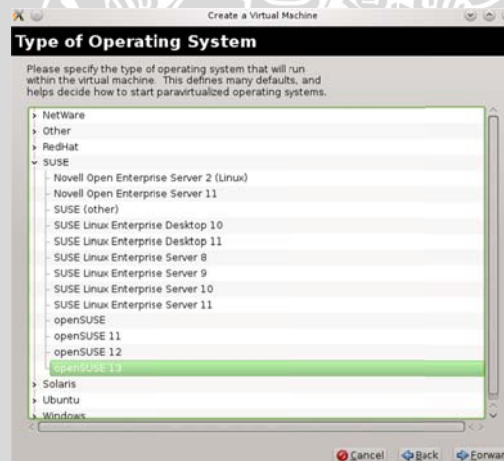
Sumber : implementasi

Setelah hypervisor terinstall, maka tahapan dilanjutkan pada pembuatan mesin virtual dan instalasi sistem operasi *guest* yang akan digunakan melalui *vm-install*.



**Gambar 4.18** Menu *vm-install* pada YasT

Sumber : implementasi



**Gambar 4.19** Pemilihan OpenSUSE 13.1 sebagai *guest OS*

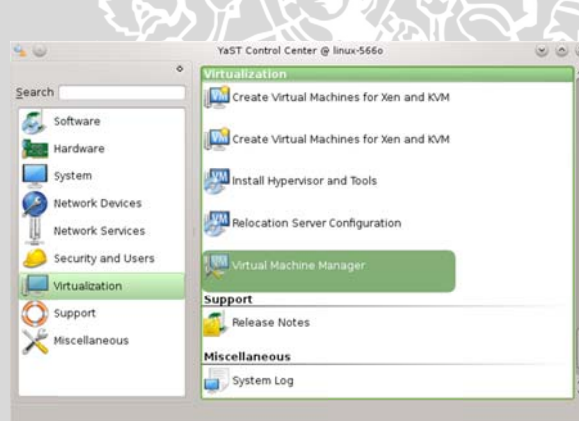
Sumber : implementasi



Gambar 4.20 Konfigurasi untuk alokasi *hardware* di dalam *guest OS*

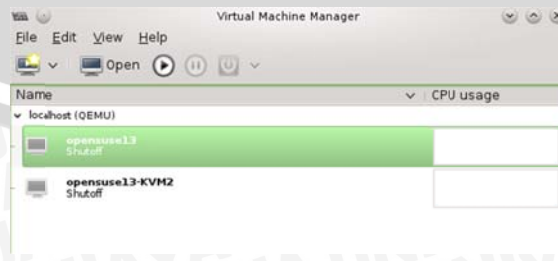
Sumber : implementasi

Mesin virtual yang telah dibuat dapat diakses melalui *YaST->Virtual Machine Manager*. Pada halaman tersebut juga memungkinkan *user* untuk melakukan konfigurasi pada mesin virtual.



Gambar 4.21 Menu *Virtual Machine Manager* pada YaST

Sumber : implementasi



Gambar 4.22 Pilihan mesin virtual yang telah dibuat pada hypervisor KVM

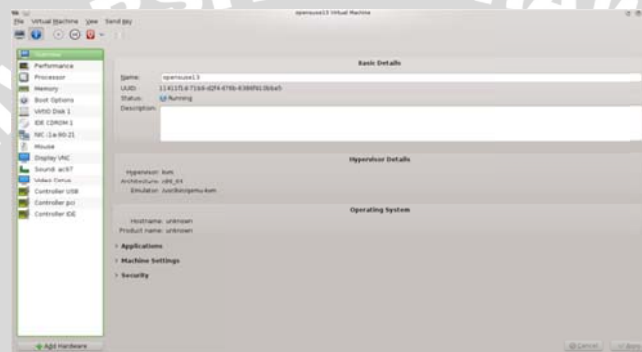
Sumber : implementasi





Gambar 4.23 Tampilan desktop OpenSUSE 13.1 pada virtual machine KVM

Sumber : implementasi



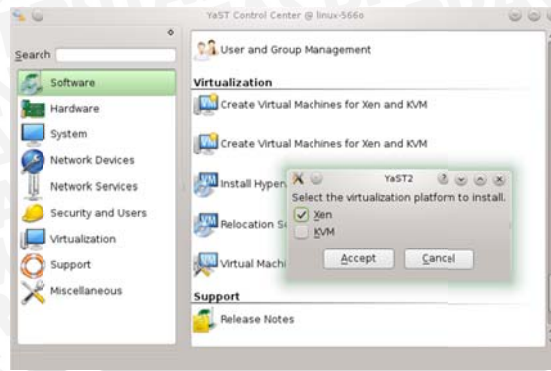
Gambar 4.24 Spesifikasi virtual machine yang dapat diubah lewat *Virtual Machine Manager*

Sumber : implementasi

#### 4.3.2.2 Instalasi XEN Untuk *Paravirtualization*

Sama halnya dengan instalasi *hypervisor* KVM, instalasi XEN juga dilakukan melalui sistem operasi *host* SUSE Linux Enterprise 11 SP 3 pada menu *Yast -> Virtualization -> Install Hypervisors and Tools*.





**Gambar 4.25** Instalasi *hypervisor* XEN pada YaST mesin *host*

Sumber : implementasi

Setelah instalasi *hypervisor* selesai, sistem akan memerintahkan *user* untuk melakukan booting ulang dan masuk ke dalam kernel XEN. Jadi *hypervisor* XEN merupakan modifikasi dari sistem operasi *host* dan berjalan pada kernel lain yang berbeda dengan sistem operasi *host*. Oleh karena itu *hypervisor* XEN disebut sebagai *bare metal hypervisor* yang langsung berjalan di atas perangkat fisik.

Modifikasi dari sistem operasi *host* diperlukan untuk menciptakan suatu lapisan *software* baru mesin virtual yang dinamakan *domain0* (dom0). Di dalam dom0 ini tertanam *drivers* yang memiliki fungsi untuk mempercepat komunikasi yang dilakukan dari mesin virtual *regular* lain (domU) ke perangkat fisik.

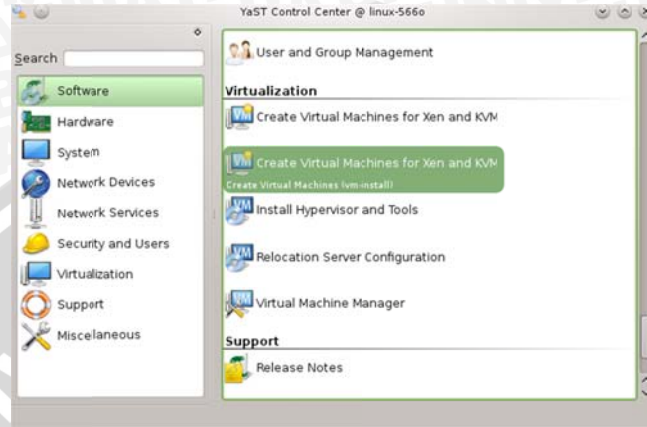


**Gambar 4.26** Tampilan desktop pada *domain0* (dom0)

Sumber : implementasi

Selain itu, dom0 memiliki hak tertinggi sehingga dapat berperan sebagai *virtual machine manager* yang bertugas menambahkan mesin virtual *regular*

(*domainU/domU*), mengatur konfigurasi di dalamnya, dan berbagi resource dari perangkat fisik (*sharing hardware*). Oleh karena itu, mesin-mesin virtual di dalam XEN memiliki kesadaran bahwa dirinya beroperasi pada lapisan virtualisasi dan mengetahui keberadaan mesin virtual lain.



**Gambar 4.27** Menu *vm-install* pada YaST *domain0* (*dom0*)

Sumber : implementasi



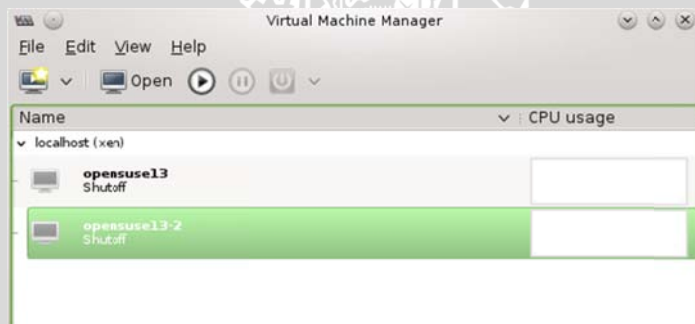
**Gambar 4.28** Pemilihan OpenSUSE 13.1 sebagai *guest OS* pada *domU*

Sumber : implementasi



Gambar 4.29 Konfigurasi untuk alokasi hardware di dalam guest OS domU

Sumber : implementasi



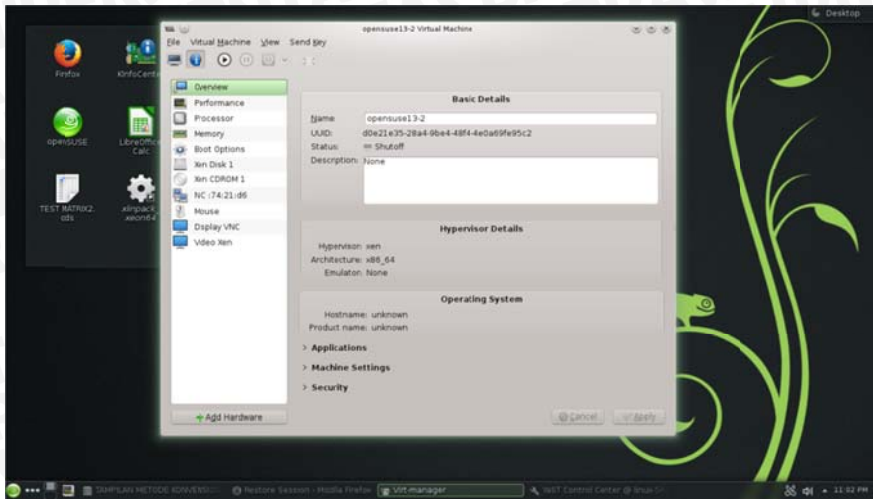
Gambar 4.30 Pilihan domU yang telah dibuat pada hypervisor XEN

Sumber : implementasi



Gambar 4.31 Tampilan desktop domU dengan sistem operasi OpenSUSE 13.1

Sumber : implementasi



Gambar 4.32 Spesifikasi domU yang dapat diubah lewat VMM pada dom0

Sumber : implementasi



